THAT WHICH IS CLAIMED IS:

1. A signal processor designed to execute variable-sized instructions that may comprise up to N elementary instruction codes, the processor comprising:

- a memory program comprising I individually addressable, parallel-connected memory banks, I being at least equal to N, wherein the codes of a program are recorded in interlaced fashion at the rate of one code per bank and per address applied to the bank,

- means for reading the program memory arranged to read a code in each of the I memory banks during a cycle for reading an instruction,

- a cycle for reading an instruction in the program memory comprising the reading of a sequence of codes that comprises the instruction code or codes to be read and can also, when the number of codes of the instruction read is smaller than I, comprise codes belonging to a following instruction.

2. A processor according to claim 1, wherein the read means comprise means for the application, to the memory banks, of the individual addresses generated from a collective value of a program counter that is incremented, before the beginning of a cycle for reading an instruction, by a value equal to the number of codes comprising the previous instruction.

3. A processor according to claim 2, wherein the means to apply addresses are laid out for the application, to each of the memory banks, of an individual read address equal to P0 or P0+1, P0 being the quotient of the division by I of the value of the program counter.

4. A processor according to claim 3, wherein the means for applying the addresses are laid out for:

- the application, to an i ranking memory bank, of an address equal to P0 when i is strictly greater than R, or for:

- the application, to an i ranking memory bank, of an address equal to P0+1 when i is smaller or equal to R,

R being the remainder of the division by I of the value of the program counter.

5. A processor according to one of the claims 1 to 4, wherein the read means comprise means for reorganizing the codes of the sequence of codes read in the program memory, according to the following algorithm:

[c'(j) = c(i),
with i = (j+R')modulo I],

in which "i" and "j" designate the ranks of the codes before and after reorganization, c(i) designates i ranking codes in their arrangement after reading in the memory, c'(j) designates j ranking codes after reorganization, R' being the remainder of the division by I of the value that was shown by the program counter during the previous clock cycle.

6. A processor according to claim 5, wherein the means for reorganizing are arranged for the application, to the codes, of the sequence of codes read in the program memory, of a circular permutation comprising a number of elementary circular permutations equal to R' or to [I-R'] depending on the direction in which the circular permutation is made.

7. A processor according to claim 6, wherein the circular permutations are performed by a barrel shifter receiving the parameter R' at a control input.

8. A processor according to one of the claims 1 to 7, wherein the read means comprise means to filter the codes that do not belong to the instruction to be read, using parallelism bits accompanying the codes.

9. A processor according to claim 8, wherein the filtered codes are replaced by no-operation codes.

10. A processor according to claim 9, wherein the code-filtering means are arranged to execute the following algorithm:

[For j = 0,
val(j=0) = "v",
s(j=0) = c'(j=0);
For j going from 1 to I,
val(j) = "v" if :
val(j-1) = "v" and if parallelism bit of c'(j)="p",
else val(j-1) = "/v" ;
s(j) = c'(j) if val(j) = "v" ;
s(j) = NOP if val(j) = "/v"],
in which val(j) is a validation term associated with each j ranking code, c'(j) capable of having two values "v" and "/v", s(j) designates j ranking outputs of the filtering means corresponding to same ranking inputs receiving a code c'(j), "NOP" is a no-operation code.

11. A processor according to one of the claims 8 to 10, wherein the non-filtered codes are sent to parallel-mounted RISC type execution units.

12. A method for the reading of variable-sized instructions that may include up to N elementary instruction codes, applicable to a signal processor, wherein the method comprises the steps consisting in :

- providing for a program memory comprising I individually addressable parallel-connected memory banks, I being at least equal to N;

- recording the codes of a program in the program memory in interlaced fashion, at a rate of one code per bank and per address applied to the bank; and,

- during a read cycle of an instruction, reading a sequence of codes in the I memory banks, said sequence comprising the code or codes of the instruction to be read and possibly also comprising, when the number of instructions codes read is smaller than I, codes belonging to a following instruction.

13. A method according to claim 12, comprising a step consisting of the application, to the memory banks, of the individual addresses generated from a collective value of a program counter, that is incremented, before the beginning of a cycle for reading an instruction, by a value equal to the number of codes contained in the previous instruction.

14. A method according to claim 13, comprising a step consisting of the application, to each to the memory banks, of an individual read address equal to P0 or P0+1, P0 being the quotient of the division by I of the value of the program counter, an i ranking memory bank receiving an address equal to P0 when i is strictly greater than R or an address equal to P0+1 when i is smaller than or equal to R, R being the remainder of the division by I of the value of the program counter.

15. A method according to one of the claims 12 to 14, comprising a step for the reorganization of the codes of the sequence of codes read in the program memory, according to the following algorithm :

[c'(j) = c(i), with i = (j+R')modulo I],

in which "i" and "j" designate the ranks of the codes before and after reorganization, c(i) designates i ranking codes in their arrangement after reading in the memory, c'(j) designates j ranking codes after reorganization, R' being the remainder of the division by I of the value that was shown by the program counter during the previous clock cycle.

16. A method according to one of the claims 12 to 15, comprising a step of filtering the codes read, that do not belong to the instruction to be read, by means of parallelism bits accompanying the codes.

17. A method according to claim 16, wherein the filtered codes are replaced by no-operation codes.

18. A method according to claim 17, wherein the codes are filtered according to the following algorithm :

[For j = 0,
val(j=0) = "v",
s(j=0) = c'(j=0);
For j going from 1 to I,
val(j) = "v" if :
val(j-1) = "v" and if parallelism bit of c'(j)="p",
else val(j-1) = "/v" ;
s(j) = c'(j) if val(j) = "v" ;
s(j) = NOP if val(j) = "/v"],

in which val(j) is a validation term associated with each j ranking code, c'(j) capable of having two values "v" and "/v", s(j) designates j ranking outputs of the filtering means corresponding to same ranking inputs receiving a code c'(j), "NOP" is a no-operation code.

19. A method according to one of the claims 16 to 18, wherein the non-filtered codes are sent to parallel-connected RISC type execution units.

Add A7